CRATOS
Center for Research on the Applications of Telematics
to Organisations and Society

The Design of Supranet Security Mechanisms

L. Delgrossi, D. Ferrari
{ldgrossi, dferrari}@pc.unicatt.it

Quaderni del CRATOS

Serie di Telematica

CTR-T98-002

Università Cattolica del Sacro Cuore – Piacenza, Italy

Abstract

This paper presents the design of a set of security mechanisms for supranets. Supranets are virtual networks - private to a group of users - that can be built on top of a physical network (e.g., the Internet) by any user of such a network making use of an appropriate software toolkit. Supranet members should feel they are part of a secure private environment from which non-members are excluded. On the other side, it may be necessary to provide the means to protect some or even all of the conversations among supranet members. This calls for the provision of a set of appropriate security mechanisms that protect supranet communications both from the outside world and within the supranet itself.

1. Introduction

This paper presents the design of a set of security mechanisms for supranets. Supranets are virtual networks - private to a group of users - that can be built on top of a physical network (e.g., the Internet) by any user of such a network making use of an appropriate software toolkit [1] [2] [3]. Supranets have been designed to offer groups of users an extended set of services: for instance, when creating a supranet, a user may fully control the topology of the network and the communication paths along which the messages are sent, restrict access to the network services to group members only, dictate rules of behaviour and impose sanctions to those members who violate these rules. In short, supranets are created, exploited, managed and deleted by their users, and this allows these users to tailor the communication environment they use to their specific needs. We feel that this ability will represent an important addition to the services offered by today's networks.

Supranet members should feel they are part of a secure private environment from which non-members are excluded. This calls for the provision of security mechanisms that protect supranet communications from the outside world. On the other side, supranet members are likely to be organised into a number of user classes, each corresponding to a predefined set of privileges; also, it may be necessary to provide the means to protect some or even all of the conversations among supranet members. This means that appropriate security mechanisms have to be provided inside a supranet as well.

The paper is organised as follows: some additional details on the design of supranets are described in Section 2; a short introduction to symmetric and asymmetric cryptography is provided in Section 3; the security mechanisms are presented in Section 4 and conclusions are summarised in Section 5.

2. Supranets

2.1 Address Space

A supranet has its own address space. Supranet addresses are assigned to supranet hosts and routers by the creator. These virtual network components are to be mapped onto physical components; a one-to-one mapping will usually be preferred for the sake of simplicity. Thus, a physical host that is also a supranet host has two addresses, a physical one and a virtual one. Furthermore, a supranet is endowed with its own name space and needs a centralised or distributed name server that translates supranet names (e.g., supranet URLs) into supranet addresses.

2.2 Supranet Packet Header

Before discussing how asymmetric encryption is applied, let us take a look at the header of a supranet packet. The supranet header includes, among others, the following relevant fields:

- *SupranetProtVers*: indicates which protocol version is being used
- *SupranetIdentifier (Sid)*: indicates which supranet the packet belongs to
- *UpperProtocol*: the identifier of the upper layer protocol
- *SendAddress*: the supranet address of the sender
- *DestAddress*: the supranet address of the destination
- *Checksum*: the checksum value calculated on the packet

- *Bitmap*: data structure that indicates a supranet user's permissions

Note that the length in bytes of some of these fields may change from supranet to supranet as it is determined by the creator's choices at supranet construction time. For example, a supranet with a strict limit of 100 on the maximum number of hosts and routers may be implemented with an 8-bit virtual address space, e.g., the SendAddress and DestAddress fields require no more than 8-bit each.

2.3 Routing

The degree of topological and connectivity control normally desired in a supranet makes it essential for all routes to be fixed. Having fixed routes in a supranet means that, for instance, to reach supranet host 'E' from supranet host 'A', supranet packets always visit supranet routers 'B', 'C' and 'D' in this order.

When the network on top of which the supranet is created is, like the Internet, connectionless at the network layer, router 'B' may be reached from host 'A' through different physical routes (hence, different Internet routers) for different packets. When, on the other hand, the underlying network is connection-oriented at the network layer (or offers both types of service, as integrated-services networks ought to [5]), also the physical route will be fixed. In all cases, the supranet need not be connection-oriented, as the appropriate static routing tables generated by the creator and stored in supranet routers will be sufficient for the purposes of topology, connectivity and multicast control.

The supranet header, is assembled by 'A' and examined by 'B', 'C', 'D' and 'E', must include the supranet addresses of the source (A) and the destination (E). In the Internet, supranet packets will travel between supranet hosts and supranet routers encapsulated into IP packets. Tunnelling is therefore an important technique in the construction of supranets.

The creator's initial specifications for all user requirements cannot be assumed to be perfect or immutable. Changes will have to be made to them during the supranet lifetime due to the addition of new members, the departure of old members, the discovery of new needs or of mistakes in the previous specifications, and so on. The creator will have to be provided with tools facilitating all reasonable modifications of the requirements on which the supranet has been based.

3. Cryptography

A number of cryptographic techniques may be adopted to protect reserved communications from attacks by unauthorised individuals. These techniques are based on the encoding of a clear text to make it unintelligible. The ciphered text is obtained by the execution of an encoding algorithm that depends on a secret key. The ciphered text is then transmitted to the receiver, who owns the key required to decipher it. In case the message is intercepted during its transmission to the destination by an intruder, he/she cannot read it unless he/she owns the secret key. Communications are secure as long as the secret key is not known by third parties, while the encoding/decoding algorithm may be public.

When the same key is used both to encrypt and decrypt a message, we say that the cryptographic system is symmetric. Symmetric cryptography is very efficient, but also has some drawbacks: first, the two parties involved in the communications need to exchange the key over a secure channel before the beginning of the transmissions; 2) a secret key is needed for each pair of interlocutors: this leads to a high number of

keys that must be utilised (for a group of N people, N*(N-1)/2 keys are needed). To solve some of these problems, asymmetric cryptographic systems have been introduced, in which the key used for the encoding is different from that for decoding.

In an asymmetric cryptographic system, each individuals has a couple of keys, a public key and a private one. The two keys have the property that each of them can be used to decipher what has been enciphered with the other. Also, knowing one of the two keys does not allow us to reconstruct the other. While the public key is made available to anybody (it is often published in public directories similar to the yellow pages), the private key is secret and only known by the individual who owns it. Asymmetric cryptography allows us to obtain a number of useful security features: for instance, to send a confidential message to a receiver, it is sufficient to encode it with his/her public key (which is always available). Only the receiver can decipher the message by using his/her private key. In the same way, a user who wishes to authenticate a message may do it by enciphering it with his/her private key: the message shall be readable for all (it is sufficient to decipher it with the correspondent public key) and certainly authentic.

With private and public keys cryptography, the number of keys that are needed is significantly reduced (2*N keys are needed for N participants) and the problem of key exchange is elegantly avoided. However, it is more complex to generate asymmetric keys and the operations of coding and decoding are more expensive in terms of execution time. For such reasons, it often happens that cryptographic systems make use of a combination of symmetric and asymmetric techniques to achieve higher efficiency.

4. Security Mechanisms

This section presents a possible approach to the design of supranet security mechanisms. This approach is based on the use of public and private keys mechanisms as described in Section 3. However, since this scheme lacks generality we expect that, in the future, it will be necessary to adapt it so that it can accommodate other encoding algorithms and single key cryptographic techniques[1]. We feel that this scheme is adequate for the purpose of the current experiments we are conducting.

The scheme makes use of cryptography and *asymmetric* keys to provide an appropriate security level. This means that, different keys are to be used for the encryption and decryption of a single message. Usually, one of these two keys is *private*, i.e., known by its owner only, and the other is *public*, i.e., known by all members in the group. Messages encrypted with a private key are to be decrypted with the correspondent public key and vice versa.

4.1 Restricted Network Access

A supranet carries all the traffic generated by the communications among its users. Only members in the group have access to the supranet, while non-members are excluded. From the security point of view, it would be desirable that the physical links used for data transmission be entirely dedicated to the traffic generated by supranet members. Potentially, this would make it more difficult for an outsider to intercept, decode, modify, and forge data packets. For example, some large

[1] To generalise the scheme a mechanism similar to the *security association* field used in the Authentication Header specifications [6] could be considered.

corporations make use of infrastructures consisting of a Private Network (PN) built with dedicated links, thus achieving a high level of protection. The drawback of this scheme is its very high cost.

Supranets, on the other hand, rely on an underlying physical network to carry the traffic generated by their users. The same physical network may simultaneously carry the traffic associated with more supranets; also, the physical network may in general carry non-supranet traffic. For example, consider a typical scenario in which a number of supranets are built on top of the Internet. Since the traffic associated with different groups is conveyed over the same physical links, it is necessary to provide the means to associate each single packet with its correspondent virtual network and to set an adequate level of protection among the different communication environments.

In our design, data packets belonging to any supranets are identified by the presence of the supranet header; the *SupranetIdentifier* (*Sid*) field included in the supranet header is used to associate each packet with its correspondent supranet. From the point of view of a supranet member, there is no difference between outsiders who are members in a different supranet and those who do not belong to any supranet: all of them are non-members with respect to which protection is needed.

To provide an adequate level of protection in this kind of environment, it would be possible to encrypt all packets travelling on supranet with a *supranet private key*. The correspondent supranet public key would be known by all supranet members, e.g., distributed off-line by the supranet creator at the time each member joins the group. Note that in no cases the *SupranetProtVers* and *Sid* fields should be encrypted, because they are necessary to identify the supranet and, with this scheme, to determine which supranet public key has to be used for the decryption. The scheme provides adequate protection among logically different communication environments. However, it has some clear disadvantages:

- the supranet public key would likely be a weak key, as it would be known by a potentially large number of users and it is used very frequently,
- the encryption and decryption of the entire packet to be executed for all packets may significantly affect overall performances, and
- not all groups and not all members in the same group need always to protect all of their messages.

Most importantly, as explained in the following sections, the scheme that we have chosen to achieve sender authentication and confidentiality within a supranet works just as well for protection against the intrusion of non-members. For all these reasons, the supranet key has not been adopted.

4.2 Confidentiality

When a message - sent by A to D - needs to be kept confidential, A must encrypt it with the public key of the receiver, in this case D ($K_{pub}(D)$). This way, only D will be able to decrypt the message by using his private key ($K_{priv}(D)$). The encryption of the payload occurs before the computation of the checksum value for the packet. In multicast conversations, group-level public and private keys can be used. This scheme protects both from external intruders and from other members of the same supranet. For this reason, we felt there was no need for additional supranet private and public keys to protect all supranet messages from non-members.

4.3 Sender Authentication

Let us now consider again the simple scenario where user A sends a message to user D along the A-B-C-D path. In a supranet, the message sent by A is encapsulated into an IP packet before it leaves host A. The same happens at supranet routers B and C. The IP packets generated at A, B, and C are different one from the others because their *SendAddress* and *DestAddress* fields indicate a different sender and destination address (A/B over the A-B path, B/C over the B-C path, and so on) and because segmentation at the IP level may be needed along some of the paths.

The Authentication Header (AH) proposed in [6] can be used to provide sender authentication at the IP layer. However, AH has a strong dependence on the IP packet structure, e.g., on the sender and destination fields of the IP header. Thus, in our scenario, it would be necessary to update at every hop not only the IP header, but also the correspondent authentication header. Most importantly, the AH scheme could be used to authenticate sender A along the A-B path, sender B along the B-C path, and so on, but it would be insufficient to provide end-to-end sender authentication, that is, to authenticate sender A to the final destination D.

For these reasons, we decided not to adopt the AH scheme and to use a mechanism based on asymmetric key cryptography instead. To authenticate his messages, sender A encrypts a portion of the supranet header with his private key ($K_{priv}(A)$). The receiver uses the public key of the sender ($K_{pub}(A)$) for the decryption. Since A is the only owner of his private key, any messages encrypted with this key must have been necessarily generated by A.

But which part of the supranet header should be encrypted with the private key of the sender? First, we have to observe that the *SendAddress* field of the supranet header should be in clear text, otherwise a destination would not know which key to use for the decryption. The *DestAddress* field of the supranet header could be encrypted with the private key of the sender, but this would not be very practical because this field is used at every intermediate host to compute the next hop on the path to the final destination. Also, since supranet addresses are virtual addresses that can be interpreted by means of appropriate tables only[2], we felt that a higher level of protection was normally not necessary in this case. So, it is sufficient for the sender to encrypt the *Checksum* field of the supranet header; this also allows us to meet the data integrity requirements.

4.4 Data Integrity

The value of the *Checksum* field of the supranet header, which is computed over the whole packet, should be encrypted with the private key of the sender. This guarantees that the contents of the packet are not manipulated after the sender has generated it and that packets are not forged by an outsider. Should the *Checksum* field not cover the whole packet, but for example only the header, it would be possible for an intruder to leave the header unmodified and substitute the payload, so that it would look like the new contents would have been written by A. With checksum covering the whole packet, even if an outsider modifies the payload and then re-computes the value of the *Checksum* field, he cannot encrypt this value with the private key of A.

[2] We assume supranet routing tables are stored in secure storage areas at each node. Should this assumption for some reasons not hold, it would be wise to encrypt the *DestAddress* field of the supranet header at the price of a presumably not very significant loss in performance.

An advantage of this scheme is that it allows the receiver automatically to detect the transmission of corrupted packets. However, a drawback is that this is normally done at a higher layer also, e.g., by the TCP protocol, thus resulting in a duplication of functions and efforts. Also, should the user explicitly ask for an unreliable service, e.g., by selecting the UDP protocol[3] at the transport layer, the supranet layer would still need to compute the value of the *Checksum* field. This would be an obvious limitation in the case of transmission of digital audio and video streams, that can tolerate a certain amount of errors, but can hardly tolerate any transmission delays. A possible alternative would be to calculate the checksum of only a portion of the payload, in the attempt to reduce the computational time required to calculate the checksum value. In this case, however, it would be necessary to impose a minimum packet size for supranet packets.

In conclusion, the detection and correction of corrupted packets are left to the higher layer protocols. At the supranet layer, for those messages that are encrypted for sender authentication purposes, a mechanism for the detection of corrupted packets is available at no extra cost. The supranet layer does not attempt to recover, e.g., by asking for packet retransmission, from the data integrity errors it detects.

4.5 Sender Anonymity

In some supranets, the sender of a message may desire to remain anonymous. A certain degree of anonymity with respect to non-members is implicitly provided by the fact that the supranet sender and destination addresses are virtual identifiers that can be related to the IP addresses of the sender and receiver only by using appropriate supranet tables, which are maintained in a secure storage area. This means that a supranet packet, even if it is intercepted by an outsider, does not contain sufficient information for the interceptor to understand who has sent it and who will receive it.

Also, it is not possible to make assumptions based on the IP addresses included into the header of the IP packet encapsulating the supranet packet, because it is in principle impossible to decide whether the sender is the actual sender of the packet or the last forwarding router, and whether the destination is the final destination or the next router. By detecting long series of supranet packets with a given supranet identifier (sid), it would be possible to guess whether a host belongs to a given supranet or not. However, the knowledge of which supranet number corresponds to a particular supranet would be required to be able to exploit this information.

To maintain anonymity within the supranet is more difficult. A possible solution is based on the introduction of an anonymity server (AS), associated with an anonymity public and private key ($K_{pub}(AS)$, $K_{priv}(AS)$). If a sender desires to send an anonymous message, he encrypts it with the anonymity public key, indicates the final destination in an appropriate field, and then sends the message to the anonymity server. The anonymity server decrypts the message by means of the anonymity private key and forwards it to the real destination after having encrypted the payload with the destination's public key. When the message is received at the destination, the address of the sender contained in the packet header is that of the anonymity server, so that the destination has no information on who the real sender of the message is.

[3] The most recent version of UDP calculates the checksum over the entire packet.

5. Conclusions

In previous works, we introduced the notion of supranets, i.e., Internet-based secure virtual networks, and described a number of potential application areas. This paper focused on the main design requirements to be considered when building security mechanism for a supranet. After the discussion of such typical networking issues as addressing, packet header format, and routing, we focused on security and showed how a set of mechanisms based on asymmetric keys cryptography can be used to address in an elegant and efficient way all the security issues that have been considered.

6. Bibliography

[1] D. Ferrari and L. Delgrossi: "*Supranets*", CRATOS Technical Report CTR-T96-001, September 1996.

[2] B. Rossi, L. Delgrossi, D. Ferrari: "*The Applications of a Toolkit for Virtual Network Creation and Management*", 4th IEEE Workshop on the Architecture and Implementation of High-Performance Communication Systems (HPCS'97), Chalkidiki, Greece, June 1997.

[3] L. Delgrossi and D. Ferrari: "*A Virtual Network Service for Integrated-Services Internetworks*", Proc. 7th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'97), St. Louis, Missouri, USA, May 1997.

[4] A. Bhimani: "*Securing the Commercial Internet*", Communications of the ACM, Vol. 39, No. 6, June 1996.

[5] D. Ferrari: "*Should an Integrated Services Internetwork be Connectionless or Connection-Oriented ?*", 6th International NOSSDAV Workshop, pp: 3-4, Zushi, Japan, April 1996.

[6] R. Atkinson: "*IP Authentication Header*", Internet RFC 1826, Proposed Standard, August 1995.

[7] R. Atkinson: "*IP Encapsulating Security Payload*", Internet RFC 1827, Proposed Standard, August 1995.

[8] S. Fotedar, M. Gerla, P. Crocetti, L. Fratta*:* "*ATM Virtual Private Networks*", Communications of the ACM, Vol. 38, N. 2, pp: 102-109, February 1995.

[9] J. M. Schneider, T. Preuss, P. S. Nielsen: "*Management of Virtual Private Networks for Integrated Broadband Communication*", Proceedings of the ACM SIGCOMM'93 Conference, pp: 224-237, September 1993.